

 Thalita MVP Jogos Notebook

(https://databricks.com)

# OBJETIVO

## Objetivo e problema

O objetivo deste trabalho é criar um pipeline de dados com foco na análise de uma base de dados sobre vendas de jogos de vídeo game. Esse pipeline permitirá a ingestão, processamento, transformação e análise do dados. Com um pipeline bem definido, é possível melhorar a eficiência, precisão e agilidade para resolver problemas reais de negócios e na obtenção de insights valiosos sobre o mercado de jogos de vídeo game. Como por exemplo, saber quais os jogos mais vendidos, por região, por plataforma, como as vendas variaram ao longo dos anos etc. Dessa forma, quero por meio deste pipeline, entender o contexto do mercado de jogos de videogame e saber prioritariamente, **quais os 10 jogos mais vendidos no intervalo temporal desta base de dados**. Pretendo também responder as seguintes perguntas:

Como são as vendas anuais por região do mundo?

Qual estilo/gênero de jogo mais vendido no mundo?

Qual Console é o campeão de vendas de jogos?

Quais Consoles com mais lançamentos?

Qual o gênero de jogo mais vendido pelo console campeão de vendas?

Qual empresa publicadora é a campeã de vendas no mercado de jogos?

# EXTRAÇÃO

## EXTRAÇÃO DE DADOS DO ARQUIVO CSV

Dataset "arquivo".

```
# Baixar o arquivo CSV para o DBFS
dbutils.fs.cp("https://raw.githubusercontent.com/Thataag/Thalita-MVP-Jogos/main/base/base.csv", "dbfs:/tmp/base.csv")
```

Out[35]: True

```
# Ler o arquivo CSV do DBFS usando Spark
arquivo = spark.read.csv("dbfs:/tmp/base.csv", header=True, inferSchema=True)
arquivo.show(100)
```

4	41 Call of Duty: Bla...	PS3 2010	Shooter	Activision	5.98	4.44	0.48	1.83	12.7
3	42 Animal Crossing: ...	DS 2005	Simulation	Nintendo	2.55	3.52	5.33	0.88	12.2
7	43  Mario Kart 7	3DS 2011	Racing	Nintendo	4.74	3.91	2.67	0.89	12.2
1	44  Halo 3	X360 2007	Shooter	Microsoft Game St...	7.97	2.83	0.13	1.21	12.1
4	45  Grand Theft Auto V	PS4 2014	Action	Take-Two Interactive	3.8	5.81	0.36	2.02	11.9
8	46 Pokemon HeartGold...	DS 2009	Action	Nintendo	4.4	2.77	3.96	0.77	11.
9	47  Super Mario 64	N64 1996	Platform	Nintendo	6.91	2.85	1.91	0.23	11.8
6	48  Gran Turismo 4	PS2 2004	Racing	Sony Computer Ent...	3.01	0.01	1.1	7.53	11.6
2	49  Super Mario Galaxy	Wii 2007	Platform	Nintendo	6.16	3.4	1.2	0.76	11.5
3	50 Pokemon Omega Rub...	3DS 2014	Role-Playing	Nintendo	4.23	3.37	3.08	0.65	11.3

# Análise Preliminar

```
from pyspark.sql.functions import count

# Mostrar um resumo estatístico dos dados numéricos
arquivo.describe().show()

# Lista de colunas categóricas
categorical_columns = ['Name', 'Platform', 'Genre', 'Publisher']

# Iterar sobre as colunas categóricas
for column in categorical_columns:
    if column in arquivo.columns:
        print(f"\nResumo da coluna categórica '{column}':")
        arquivo.groupBy(column).agg(count('*').alias('Count')).show()
    else:
        print(f"Coluna '{column}' não encontrada no DataFrame.")
```

Resumo da coluna categórica 'Publisher':	
	Publisher Count
	Funbox Media  6
	Media Rings  3
	Iceberg Interactive  3
	Tigervision  3
	bitComposer Games  5
	3DO  36
	Telegames  8
	Jack of All Games  3
	Nihon Falcom Corp...  7
	Sting  9
	id Software  1
	IE Institute  5
	Game Life  2
	Karin Entertainment  1

```
# Visualizando o DataFrame
arquivo.show()

# Obtendo informações do esquema do DataFrame
arquivo.printSchema()

# Obtendo os tipos das colunas
print(arquivo.dtypes)
```

8	Wii Play	Wii 2006	Misc	Nintendo	14.03	9.2	2.93	2.85	29.0
9	New Super Mario B...	Wii 2009	Platform	Nintendo	14.59	7.06	4.7	2.26	28.6
10	Duck Hunt	NES 1984	Shooter	Nintendo	26.93	0.63	0.28	0.47	28.3
11	Nintendogs	DS 2005	Simulation	Nintendo	9.07	11.0	1.93	2.75	24.7
12	Mario Kart DS	DS 2005	Racing	Nintendo	9.81	7.57	4.13	1.92	23.4
13	Pokemon Gold/Poke...	GB 1999	Role-Playing	Nintendo	9.0	6.18	7.2	0.71	23.
14	Wii Fit	Wii 2007	Sports	Nintendo	8.94	8.03	3.6	2.15	22.7
15	Wii Fit Plus	Wii 2009	Sports	Nintendo	9.09	8.59	2.53	1.79	22.
16	Kinect Adventures!	X360 2010	Misc Microsoft Game St...	14.97	4.94	0.24	1.67	21.8	
17	Grand Theft Auto V	PS3 2013	Action Take-Two Interactive	7.01	9.27	0.97	4.14	21.	
18	Grand Theft Auto:...	PS2 2004	Action Take-Two Interactive	9.43	0.4	0.41	10.57	20.8	

```
# Exibir o schema do DataFrame
arquivo.printSchema()

# Exibir o número de linhas e colunas
num_rows = arquivo.count()
num_cols = len(arquivo.columns)
print(f"Número de linhas: {num_rows}")
print(f"Número de colunas: {num_cols}")

# Exibir o tipo de dados de cada coluna
print("Tipos de dados de cada coluna:")
for column, dtype in arquivo.dtypes:
    print(f"{column}: {dtype}")

|-- Publisher: string (nullable = true)
|-- NA_Sales: double (nullable = true)
|-- EU_Sales: double (nullable = true)
|-- JP_Sales: double (nullable = true)
|-- Other_Sales: double (nullable = true)
|-- Global_Sales: double (nullable = true)

Número de linhas: 16598
Número de colunas: 11
Tipos de dados de cada coluna:
Rank: int
Name: string
Platform: string
Year: string
Genre: string
Publisher: string
NA_Sales: double
EU_Sales: double
JP_Sales: double
Other_Sales: double
Global_Sales: double
```

## MODELAGEM

### FLAT TABLE

O dataset foi baixado no Kaggle e é composto de 16.598 linhas e 11 colunas. Segue abaixo, o conteúdo de cada coluna:

Rank: Posição do jogo no ranking de vendas;

Name: Nome do jogo;

Platform: Console que lançou o jogo;

Year: Ano em que o jogo foi lançado;

Genre: Gênero/Estilo do jogo;

Publisher: Empresa responsável pela publicação do jogo;

NA\_Sales: Vendas na América do Norte, em milhões de dólares;

EU\_Sales: Vendas na Europa, em milhões de dólares;

JP\_Sales: Vendas no Japão, em milhões de dólares;

Other\_Sales: Vendas em outras regiões, em milhões de dólares;

Global\_Sales: Venda total no mundo.

.....

As variáveis que pertencem à categoria quantitativa (numéricas) são: Rank, Year, NA\_Sales, EU\_Sales, JP\_Sales, Other\_Sales and Global\_Sales.

As variáveis classificadas como qualitativas (categóricas) do conjunto de dados em estudo são: Name, Platform, Genre and Publisher.

.....  
Domínios dos Dados Desejado:

- Rank: Valores inteiros representando a posição do jogo no ranking de vendas. **Valor mín: 1 e máx: 16600**
- Name: Strings não vazias com nomes dos jogos.

Resumo da coluna categórica 'Name':

Need for Speed: Most Wanted 12

Ratatouille 9

FIFA 14 9

LEGO Marvel Super Heroes 9

Madden NFL 07 9

...

Ar tonelico Qoga: Knell of Ar Ciel 1

Galaga: Destination Earth 1

Nintendo Presents: Crossword Collection 1

TrackMania: Build to Race 1

Know How 2 1

Name: count, **Length: 11493**.

- Platform: Strings representando plataformas (ex.: 'PS4', 'XOne', 'Switch').

Resumo da coluna categórica 'Platform':

DS 2163

PS2 2161

PS3 1329

Wii 1325

X360 1265

PSP 1213

PS 1196

PC 960

XB 824

GBA 822

GC 556

3DS 509

PSV 413

PS4 336

N64 319

SNES 239

XOne 213

SAT 173

WiiU 143

2600 133

NES 98

GB 98

DC 52

GEN 27

NG 12

SCD 6

WS 6

3DO 3

TG16 2

GG 1

PCFX 1 Name: count.

- Year: Valores inteiros representando o ano (ex.: 2019, 2020, 2021). **Valor mín:1980 e máx: 2020**

- Genre: Strings representando gêneros de jogos (ex.: 'Ação', 'Aventura', 'RPG').

Resumo da coluna categórica 'Genre':

Action 3316

Sports 2346

Misc 1739

Role-Playing 1488

Shooter 1310

Adventure 1286

Racing 1249

Platform 886

Simulation 867

Fighting 848

Strategy 681

Puzzle 582

Name: count.

- Publisher: Strings com nomes das editoras/Publicadoras (ex.: 'Nintendo', 'EA').

Resumo da coluna categórica 'Publisher':

Electronic Arts 1351

Activision 975

Namco Bandai Games 932

Ubisoft 921

Konami Digital Entertainment 832

...

Warp 1

New 1

Elite 1

Evolution Games 1

UIG Entertainment 1

Name: count, **Length: 578**.

- NA\_Sales: Valores decimais positivos representando as vendas na América do Norte. **Valor mín: 0 e máx: 41,49**
- EU\_Sales: Valores decimais positivos representando as vendas na Europa. **Valor mín: 0 e máx: 29,02**
- JP\_Sales: Valores decimais positivos representando as vendas no Japão. **Valor mín: 0 e máx: 10,22**
- Other\_Sales: Valores decimais positivos representando as vendas em outras regiões. **Valor mín: 0 e máx: 10,57**
- Global\_Sales: Valores decimais positivos representando as vendas globais, que é a soma das vendas regionais. **Valor mín: 0 e máx: 82,74**

## TRANSFORMAÇÃO

INICIANDO A TRANSFORMAÇÃO DOS DADOS:

```
from pyspark.sql.functions import col, isnan, count, when

# Função para contar valores nulos em cada coluna
def contar_valores_nulos(df):
    # Contagem de valores nulos para cada coluna
    nulos = df.select([count(when(col(c).contains('None') | col(c).contains('NULL') | col(c).contains('N/A') | (col(c) == '') | col(c).isnull())) for c in df.columns])

    return nulos

# Exemplo de uso
nulos_df = contar_valores_nulos(arquivo)
nulos_df.show()
```

Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
0	3	0	271	0	59	0	0	0	0	0

Não há muitos valores faltantes. Não estamos perdendo muita informação. Já é um bom conjunto de dados pronto.

Contudo, As colunas 'Year' e 'Publisher' apresentam 271 e 59 valores sem registro, respectivamente. Como irei fazer alguns estudos onde o ano de lançamento será uma informação importante, vou eliminar dados cujo ano não seja informado. E as publicadoras sem registro, classificarei como 'unknown'.

```
from pyspark.sql import SparkSession

# Iniciar uma sessão do Spark
spark = SparkSession.builder.appName("DataCleaning").getOrCreate()

# Carregar os dados em um DataFrame do Spark
arquivo = spark.read.csv("dbfs:/tmp/base.csv", header=True, inferSchema=True)

# Removendo dados que não têm o ano de lançamento informado
arquivo = arquivo.dropna(subset=['Year'])

# Registrando como 'unknown' publicadoras faltantes no dataset
arquivo = arquivo.fillna({'Publisher': 'unknown'})

from pyspark.sql.functions import col, count, when

# Validação dos Dados. Verificando novamente se ainda há valores nulos no dataset
arquivo.select([count(when(col(c).isNull(), c)).alias(c) for c in arquivo.columns]).show()
```

Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
0	0	0	0	0	0	0	0	0	0	0

Agora, sem valores nulos! Vou agora para outros tratamentos necessários: A coluna Year está como string, vou transformar para int.

```
# Transformando Ano de Lançamento em Int
arquivo = arquivo.withColumn('Year', col('Year').cast('int'))

# Visualizando o DataFrame
arquivo.show()
```

1	Wii Sports	Wii	2006	Sports	Nintendo	41.49	29.02	3.77	8.46	82.7
2	Super Mario Bros.	NES	1985	Platform	Nintendo	29.08	3.58	6.81	0.77	40.2
3	Mario Kart Wii	Wii	2008	Racing	Nintendo	15.85	12.88	3.79	3.31	35.8
4	Wii Sports Resort	Wii	2009	Sports	Nintendo	15.75	11.01	3.28	2.96	33.
5	Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	Nintendo	11.27	8.89	10.22	1.0	31.3
6	Tetris	GB	1989	Puzzle	Nintendo	23.2	2.26	4.22	0.58	30.2
7	New Super Mario Bros. U	DS	2006	Platform	Nintendo	11.38	9.23	6.5	2.9	30.0
8	Wii Play	Wii	2006	Misc	Nintendo	14.03	9.2	2.93	2.85	29.0
9	New Super Mario Bros. U	Wii	2009	Platform	Nintendo	14.59	7.06	4.7	2.26	28.6
10	Duck Hunt	NES	1984	Shooter	Nintendo	26.93	0.63	0.28	0.47	28.3

```
# Visualizando dataframe
print(arquivo.dtypes)
print(arquivo.head())

[(('Rank', 'int'), ('Name', 'string'), ('Platform', 'string'), ('Year', 'int'), ('Genre', 'string'), ('Publisher', 'string'), ('NA_Sales', 'double'), ('EU_Sales', 'double'), ('JP_Sales', 'double'), ('Other_Sales', 'double'), ('Global_Sales', 'double'))]
Row(Rank=1, Name='Wii Sports', Platform='Wii', Year=2006, Genre='Sports', Publisher='Nintendo', NA_Sales=41.49, EU_Sales=29.02, JP_Sales=3.77, Other_Sales=8.46, Global_Sales=82.74)
```

# Analisando o período estudado no dataset:

```
# Quantidade de lançamentos por ano
lancamentos_por_ano = arquivo.groupBy('Year').count().orderBy('Year')

# Coletar todos os registros
todos_lancamentos_por_ano = lancamentos_por_ano.collect()

# Visualizando todos os resultados
for linha in todos_lancamentos_por_ano:
    print(f"Ano: {linha['Year']}, Lançamentos: {linha['count']}")

Ano: 1998, Lançamentos: 379
Ano: 1999, Lançamentos: 338
Ano: 2000, Lançamentos: 349
Ano: 2001, Lançamentos: 482
Ano: 2002, Lançamentos: 829
Ano: 2003, Lançamentos: 775
Ano: 2004, Lançamentos: 763
Ano: 2005, Lançamentos: 941
Ano: 2006, Lançamentos: 1008
Ano: 2007, Lançamentos: 1202
Ano: 2008, Lançamentos: 1428
Ano: 2009, Lançamentos: 1431
Ano: 2010, Lançamentos: 1259
Ano: 2011, Lançamentos: 1139
Ano: 2012, Lançamentos: 657
Ano: 2013, Lançamentos: 546
Ano: 2014, Lançamentos: 582
Ano: 2015, Lançamentos: 614
Ano: 2016, Lançamentos: 344
Ano: 2017, Lançamentos: 3
Ano: 2020, Lançamentos: 1
```

O período total é de 1980 a 2020. Porém, como os anos de 2020 e 2017 apresentam dados faltantes e os anos de 2018 e 2019 não aparecem na lista, considero então que os lançamentos após o ano de 2016 são incompletos e podem atrapalhar este estudo. Dessa forma, opto por remover estes lançamentos do dataset.

```
# Eliminando todos os dados cujo ano de lançamento seja maior ou igual a 2017
arquivo = arquivo.filter(arquivo['Year'] < 2017)

# Visualizando os anos contidos no dataset
anos_unicos = arquivo.select('Year').distinct().orderBy('Year')
todos_anos_unicos = anos_unicos.collect()

# Imprimindo todos os anos únicos
for linha in todos_anos_unicos:
    print(f"Ano: {linha['Year']}")
```

```
Ano: 1996
Ano: 1997
Ano: 1998
```

Ano: 2015  
Ano: 2016

Agora, o período estudado é de 1980 a 2016.

No dataset já consta a coluna 'Rank', correspondendo ao rank de vendas totais de jogos por console. Entretanto, para saber realmente quais os jogos mais vendidos, temos que considerar que muitos jogos são multiplataforma, ou seja, lançadas por consoles diversos.

```
from pyspark.sql.functions import col

# Observando 20 principais jogos lançados para mais plataformas
top_20_jogos = (arquivo.groupBy('Name')
                 .count()
                 .orderBy(col('count').desc())
                 .limit(20))

# Visualizando os 20 principais jogos
top_20_jogos.show()
```

LEGO Marvel Super...	9
Ratatouille	9
FIFA 14	9
Terraria	8
Monopoly	8
Madden NFL 08	8
Lego Batman 3: Be...	8
Angry Birds Star ...	8
FIFA Soccer 13	8
Madden NFL 07	8
LEGO Jurassic World	8
LEGO The Hobbit	8
FIFA 15	8
LEGO Star Wars II...	8
Cars	8
The LEGO Movie Vi...	8
Need for Speed Ca...	7
Star Wars The Clo...	7
Spider-Man 3	7
-----+-----+	

Como exemplo, vou consultar as vendas do Jogo Ratatouille que foi lançado em 9 consoles diferentes

```
# Filtrando registros onde o nome do jogo é 'Ratatouille' e ordenando pelas vendas globais
ratatouille_sales = (arquivo.filter(col('Name') == 'Ratatouille')
                      .orderBy('Global_Sales', ascending=True))

# Visualizando os resultados
ratatouille_sales.show()
```

Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
14398 Ratatouille	PC	2007	Action	THQ	0.01	0.01	0.0	0.0	0.03	
9027 Ratatouille	GC	2007	Action	THQ	0.11	0.03	0.0	0.0	0.14	
7705 Ratatouille	GBA	2007	Action	THQ	0.14	0.05	0.0	0.0	0.2	
6385 Ratatouille	X360	2007	Action	THQ	0.23	0.02	0.0	0.02	0.27	
3853 Ratatouille	PS3	2007	Action	THQ	0.09	0.32	0.0	0.11	0.52	
3859 Ratatouille	Wii	2007	Action	THQ	0.44	0.04	0.0	0.04	0.52	
3031 Ratatouille	PSP	2007	Action	THQ	0.22	0.28	0.0	0.16	0.67	
2447 Ratatouille	PS2	2007	Action	THQ	0.31	0.0	0.0	0.53	0.85	
1580 Ratatouille	DS	2007	Action	THQ	0.5	0.62	0.0	0.14	1.26	

Para contabilizar a venda deste jogo especificamente e de todos os demais jogos multiplataforma e posicioná-los no Rank, devemos somar as vendas totais desses jogos em cada console que o comercializou.

```
from pyspark.sql.functions import sum

# Agrupando jogos pela soma de vendas total
ranking_vendas = (arquivo.groupBy('Name')
    .agg(sum('Global_Sales').alias('Total_Sales'))
    .orderBy('Total_Sales', ascending=False)
    .limit(10))
```

Após o tratamento dos dados realizado acima, podemos visualizar finalmente, os dez jogos mais vendidos entre 1980 e 2016, são eles:

```
# Visualizando o ranking dos 10 jogos mais vendidos
ranking_vendas.show()
```

Name	Total_Sales
Wii Sports	82.74
Grand Theft Auto V	55.92
Super Mario Bros.	45.31
Tetris	35.84
Mario Kart Wii	35.82
Wii Sports Resort	33.0
Pokemon Red/Pokemon Blue	31.37
Call of Duty: Modern Warfare	30.83
New Super Mario Bros. U	30.01
Call of Duty: Black Ops II	29.72

## FINALIZANDO A TRANSFORMAÇÃO DOS DADOS

### CARGA

Carregando os dados transformados para um banco de dados SQL.

```

import requests

# Obter o endereço IP público
ip = requests.get('https://api.ipify.org').text
print(f'Endereço IP público do Databricks: {ip}')


# Configurar os parâmetros de conexão ao PostgreSQL
jdbc_url = "jdbc:postgresql://34.170.76.253:5432/db"
connection_properties = {
    "user": "thalita",
    "password": "guy182",
    "driver": "org.postgresql.Driver",
    "batchsize": "1000", # Ajustar o tamanho do lote
    "isolationLevel": "NONE", # Ajustar o nível de isolamento se necessário
}

# Nome da tabela no PostgreSQL
tabela_destino = "mvp_jogos"

arquivo = arquivo.repartition(1)

# Escrever o DataFrame no PostgreSQL
arquivo.write.jdbc(
    url=jdbc_url,
    table=tabela_destino,
    mode="append", # ou "overwrite" se você quiser substituir os dados existentes
    properties=connection_properties
)

```

Endereço IP público do Databricks: 35.93.247.38

## ANÁLISE DA QUALIDADE DE DADOS:

A análise da qualidade dos dados é muito importante, pois a confiabilidade e a precisão das análises dependem diretamente da integridade dos dados utilizados. Neste pequeno relatório, apresento a análise de qualidade realizada sobre o conjunto de dados fornecido, que já veio praticamente pronto e necessitou de poucos ajustes. O conjunto de dados apresentado contém informações sobre vendas de jogos de videogame e os principais atributos incluídos são:

- \*'Name': Nome do jogo
- \*'Year': Ano de lançamento de jogo
- \*'Platform': Console/Plataforma
- \*'Genre': Gênero do jogo
- \*'Publisher': Empresa publicadora dos jogos
- \*'Global\_Sales': Total de Vendas Globais

Além de conter também, atributos referentes à vendas por Região do Globo.

.....

**Valores Faltantes** A verificação inicial de valores faltantes revelou que a maioria dos atributos já estava completa, com poucos registros apresentando valores ausentes. E os ajustes realizados foram: Eliminação dos dados cujo ano não foi apresentado na tabela 'Year' e preenchimento dos campos sem registro na tabela 'Publisher' como "Unknown"

**Consistência de dados** Foi verificado se os dados estavam em formatos consistentes e padronizados e não foi necessária alteração no tipo.

**Identificação de Outliers** Foi identificado no intervalo temporário da base utilizada, que os anos de 2020 e 2017 apresentaram dados faltantes e os anos de 2018 e 2019 não apareceram na lista. Removi portanto, os anos posteriores a 2016 por serem ruidosos e incompletos, podendo comprometer este estudo.

Em resumo, o conjunto de dados já estava praticamente pronto e exigiu apenas ajustes mínimos. Após a análise e as correções efetuadas, os dados ficaram completos, consistentes e prontos para serem utilizados nas análises subsequentes. Este nível de qualidade inicial indica que os dados foram bem curados e preparados antes de serem disponibilizados, o que facilita o trabalho de análise e aumenta a confiança nos resultados obtidos.

### Análises para resposta às perguntas do problema.

Para geração de gráficos utilizei biblioteca Matplotlib e para as demais consultas e visualizações, Spark. Vou primeiramente, explorar os dados por Região. Quero o total de vendas anuais por região.

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, sum
import matplotlib.pyplot as plt

# Iniciar uma sessão do Spark
spark = SparkSession.builder.appName("SalesAnalysis").getOrCreate()

# Carregar os dados em um DataFrame do Spark
arquivo = spark.read.csv("dbfs:/tmp/base.csv", header=True, inferSchema=True)

# Agrupando por ano e calculando as vendas por região
region_yearly_sales = (arquivo.groupBy('Year')
                        .agg(sum('NA_Sales').alias('Total_NA_Sales'),
                             sum('EU_Sales').alias('Total_EU_Sales'),
                             sum('JP_Sales').alias('Total_JP_Sales'),
                             sum('Other_Sales').alias('Total_Other_Sales'))
                        .orderBy('Year'))

# Mostrando a tabela de vendas anuais por região
region_yearly_sales.show()

# Coletando os dados para plotagem
data = region_yearly_sales.collect()

# Extraíndo os dados para listas separadas
years = [row['Year'] for row in data]
total_na_sales = [row['Total_NA_Sales'] for row in data]
total_eu_sales = [row['Total_EU_Sales'] for row in data]
total_jp_sales = [row['Total_JP_Sales'] for row in data]
total_other_sales = [row['Total_Other_Sales'] for row in data]

# Plotando os dados
plt.figure(figsize=(10, 6))

plt.plot(years, total_na_sales, marker='o', label='NA Sales')
plt.plot(years, total_eu_sales, marker='o', label='EU Sales')
plt.plot(years, total_jp_sales, marker='o', label='JP Sales')
plt.plot(years, total_other_sales, marker='o', label='Other Sales')

plt.title('Total Sales by Region per Year')
plt.xlabel('Year')
plt.ylabel('Total Sales')
plt.legend()
plt.grid(True)
plt.show()
```

Command skipped

Em seguida, vou verificar os gêneros mais vendidos

```
# Agrupando por gênero e calculando as vendas globais
genre_sales = (arquivo.groupBy('Genre')
                 .agg(sum('Global_Sales').alias('Total_Global_Sales'))
                 .orderBy(col('Total_Global_Sales').desc()))

# Mostrando a tabela dos gêneros mais vendidos
genre_sales.show()

# Coletando os dados para plotagem
data = genre_sales.collect()

# Extraíndo os dados para listas separadas
genres = [row['Genre'] for row in data]
total_sales = [row['Total_Global_Sales'] for row in data]

# Plotando os dados
plt.figure(figsize=(12, 8))
plt.barh(genres, total_sales, color='skyblue')
plt.xlabel('Total Global Sales')
plt.ylabel('Genre')
plt.title('Total Global Sales by Genre')
plt.gca().invert_yaxis() # Inverter o eixo y para mostrar os maiores valores no topo
plt.grid(axis='x')
plt.show()
```

Command skipped

Agora vou analisar os consoles com mais números de vendas e com mais lançamentos de jogos.

```
# Agrupando por plataforma e calculando as vendas globais
platform_sales = (arquivo.groupBy('Platform')
                  .agg(sum('Global_Sales').alias('Total_Global_Sales'))
                  .orderBy(col('Total_Global_Sales').desc())
                  .limit(10))

# Mostrando a tabela das plataformas que mais venderam jogos
platform_sales.show()

# Coletando os dados para plotagem
data = platform_sales.collect()

# Extraíndo os dados para listas separadas
platforms = [row['Platform'] for row in data]
total_sales = [row['Total_Global_Sales'] for row in data]

# Plotando os dados
plt.figure(figsize=(12, 8))
plt.barh(platforms, total_sales, color='skyblue')
plt.xlabel('Total Global Sales')
plt.ylabel('Platform')
plt.title('Total Global Sales by Platform (Top 10)')
plt.gca().invert_yaxis() # Inverter o eixo y para mostrar os maiores valores no topo
plt.grid(axis='x')
plt.show()
```

Command skipped

```
# Contando o número de lançamentos por plataforma
platform_releases = (arquivo.groupBy('Platform')
    .agg(count('Platform').alias('Lançamentos'))
    .orderBy(col('Lançamentos').desc())
    .limit(10))

# Mostrando a tabela das plataformas que mais tiveram lançamentos
platform_releases.show()

# Coletando os dados para plotagem (opcional, dependendo da necessidade)
data = platform_releases.collect()

# Extrair os dados para listas separadas (opcional, dependendo da necessidade)
platforms = [row['Platform'] for row in data]
num_releases = [row['Lançamentos'] for row in data]
```

Command skipped

Como podemos observar acima, o PS2 é o console mais vendido. E de fato, é o console de videogame mais vendido e o mais popular de todos os tempos. Foi lançado no dia 4 de março de 2000 no Japão, no dia 26 de outubro na América do Norte, e posteriormente, no dia 24 de novembro na Europa. É o sucessor do PS (PlayStation). **Por conta de sua importância, podemos explorar melhor os seus dados, pesquisando quais os 10 jogos do PS2 foram os mais vendidos e quais os gêneros dos jogos mais vendidos para este Console:**

```
# Filtrando os jogos da plataforma PS2
ps2_games = arquivo.filter(col('Platform') == 'PS2')

# Selecionando os campos de interesse e ordenando pelas vendas globais
ps2_top10 = (ps2_games.select('Name', 'Global_Sales')
    .orderBy('Global_Sales', ascending=False)
    .limit(10))

# Mostrando a tabela dos jogos mais vendidos no PS2
ps2_top10.show()
```

Command skipped

```
# Filtrando os jogos da plataforma PS2
ps2_games = arquivo.filter(col('Platform') == 'PS2')

# Agrupando por gênero e somando as vendas globais
popular_genres_ps2 = (ps2_games.groupBy('Genre')
    .agg(sum('Global_Sales').alias('Total_Global_Sales'))
    .orderBy('Total_Global_Sales', ascending=False))

# Mostrando a tabela dos gêneros mais vendidos no PS2
popular_genres_ps2.show()
```

Command skipped

Agora voltando aos dedos gerais, vou explorar as publicadoras. Quero saber quais delas mais lançaram jogos e as que mais venderam jogos no intervalo de tempo do conjunto de dados:

```
# Contando o número de lançamentos por publicadora
most_releases_by_publisher = (arquivo.groupBy('Publisher')
    .agg(count('*').alias('Lançamentos'))
    .orderBy(col('Lançamentos').desc())
    .limit(10))

# Mostrando o resultado
most_releases_by_publisher.show()
```

Command skipped

```
# Organizando as publicadoras pelo número de vendas globais
most_selling_publishers = (arquivo.groupBy('Publisher')
                            .agg(sum('Global_Sales').alias('Total_Global_Sales'))
                            .orderBy(col('Total_Global_Sales').desc())
                            .limit(10))

# Mostrando o resultado
most_selling_publishers.show()
```

Command skipped

## Autoavaliação

Considero que meu trabalho foi simples e essa atividade foi bastante desafiadora pra mim, tendo em vista que essa foi minha primeira pipeline de dados e primeiro contato com o Databricks e com Engenharia de dados, contudo, acredito que cumpri o que foi proposto como atividade final do módulo.

Optei por utilizar a linguagem python porque me dá mais segurança.

O trabalho acima foi realizado no intuito de construir um pipeline de dados a partir de um dataset sobre vendas de videogames durante o período de 1980 à 2016, é um tema que me atrai e considero interessante.

Baixei do Kaggle e após a extração e carga no Databricks, explorei a base para entender sua composição e produzir o catálogo de dados/modelagem. Em seguida comecei a tratar os dados.

No processo de transformação, foi identificado os registros nulos nas tabelas "Year" e 'Publisher', onde optei por excluir dados cujo ano não foi informado e classifiquei como unknown onde não havia registro na coluna 'Publisher'.

Alterei o tipo da coluna 'Year' de string para int.

Analisei o período do dataset e observei que os anos de 2020 e 2017 apresentavam dados faltantes e os anos de 2018 e 2019 não apareciam na lista, então considerei que os lançamentos após o ano de 2016 eram incompletos/ruidosos e poderiam atrapalhar este estudo. Dessa forma, optei por remover estes lançamentos do dataset, definindo o período de 1980 a 2016.

Nessa etapa, observei também que, a coluna 'Rank' não estava considerando os jogos multiplataforma, ou seja, aqueles jogos que foram comercializados para e por diferentes consoles. Para contabilizar a venda dos jogos multiplataforma e visualizá-los num rank, somei as vendas totais desses jogos em cada console que o comercializou. Após isso, pude visualizar os dez jogos mais vendidos entre 1980 e 2016 em resposta ao meu problema e principal pergunta a ser respondida neste estudo.

Após isso fiz uma análise sobre a qualidade dos dados e, finalizei com as análises e proporcionaram respostas à todas as demais perguntas do problema além de mais algumas que surgiram ao longo da análise e considerações a respeito do mercado de jogos de videogames, descritas na conclusão e solução do problema.

## Conclusão e Solução do problema:

Após as etapas de ETL, pude deixar a base em condições para que eu pudesse realizar as análises que responderiam minha perguntas e solução do meu problema. Por meio deste trabalho, foi possível extrair informações relevantes sobre a indústria de jogos eletrônicos no mundo e sobre as tendências e padrões do mercado global de videogames. Ao examinar os dados, conseguimos identificar várias características importantes que ajudam a compreender melhor a dinâmica deste setor.

### **Os 10 jogos mais vendidos considerando os jogos multiplataforma foram:**

Wii Sports

Grand Theft Auto V

Super Mario Bros.

Tetris

Mario Kart Wii

Wii Sports Resort

Pokemon Red/Pokem...

Call of Duty: Mod...

New Super Mario B...

Call of Duty: Bla...

**Vendas por Região:** As regiões da América do Norte, Europa, Japão e outras partes do mundo apresentaram padrões distintos de vendas ao longo dos anos. A América do Norte e a Europa dominaram as vendas globais.

**Gêneros de jogos mais vendidos no mundo:** Gêneros como ação, esporte e tiro lideraram as vendas globais. Esta análise ajudou a destacar quais tipos de jogos têm maior apelo no mercado.

**Console campeão de vendas:** PS2.

**Os 10 jogos mais vendidos de PS2 foram:**

Grand Theft Auto

Grand Theft Auto II

Gran Turismo 3

Grand Theft Auto III

Gran Turismo 4

Final Fantasy X

Need for Speed

Need for Speed

Medal of Honor

Kingdom Hearts

**Os gêneros de jogos mais vendidos de PS2 foram:**

Sports

Action

Racing

Shooter

Misc

Role-Playing

Fighting

Platform

Simulation

Adventure

Strategy

Puzzle

**Console com mais lançamentos:** DS.

As plataformas que mais venderam jogos ao longo dos anos foram identificadas, com destaque para consoles icônicos como o PlayStation 2, Xbox 360 e Nintendo Wii. Estes consoles não apenas tiveram um alto volume de vendas, mas também um número significativo de lançamentos. A análise dos lançamentos por plataforma revelou que certas plataformas têm um número significativamente maior de jogos lançados, o que pode influenciar a popularidade e as vendas desses consoles.

**Publicadora campeã de vendas:** Nintendo.

**Publicadora com mais lançamentos:** Electronic Arts

As publicadoras que mais venderam jogos, como Nintendo, Electronic Arts e Activision, foram identificadas, demonstrando a importância dessas empresas no mercado global de videogames. Elas não apenas lideram em vendas, mas também têm um grande número de lançamentos.

## Implicações e Recomendações

- **Estratégias de Marketing Regionalizadas:** As diferenças nas vendas por região sugerem que as estratégias de marketing e distribuição devem ser adaptadas para atender às preferências regionais. Campanhas focadas em gêneros populares em cada região podem aumentar a eficácia.
- **Foco em Consoles e Gêneros Populares:** Investir no desenvolvimento de jogos para as plataformas mais vendidas e nos gêneros mais populares pode maximizar o retorno sobre o investimento para desenvolvedores e publicadoras.
- **Análise de Tendências Futuras:** Continuar a monitorar as tendências de vendas pode ajudar a prever mudanças no mercado e ajustar as estratégias de desenvolvimento e marketing conforme necessário.

Limitações e Trabalho Futuro

Embora a análise tenha fornecido insights valiosos, algumas limitações devem ser consideradas. A base de dados pode não incluir todas as vendas ou lançamentos, e os dados históricos podem não refletir completamente as tendências atuais. Trabalhos futuros podem incluir a análise de dados adicionais, como vendas digitais e mobile, para obter uma visão mais abrangente do mercado.